Serving Ocean Model Data on the Cloud

Michael Meisinger[1], Claudiu Farcas[1], Emilia Farcas[1], Charles Alexander[2], Matthew Arrott[1], Jeff de La Beaujardière[2], Paul Hubbard[21], Roy Mendelssohn[3], and Richard Signell[4]
[1] Calit2, University of California at San Diego, La Jolla, CA 92093-0436, USA
[2] NOAA IOOS Program Office, 1100 Wayne Ave Suite 1225, Silver Spring, MD 20910, USA
[3] U.S. Geological Survey, 384 Woods Hole Road, Woods Hole, MA 02543-1598, USA
[4] NOAA/NMFS/SWFSC/ ERD, Pacific Grove, CA, USA

***Abstract*- The NOAA-led Integrated Ocean Observing System (IOOS) and the NSF-funded Ocean Observatories Initiative Cyberinfrastructure Project (OOI-CI) are collaborating on a prototype data delivery system for numerical model output and other gridded data using cloud computing. The strategy is to take an existing distributed system for delivering gridded data and redeploy on the cloud, making modifications to the system that allow it to harness the scalability of the cloud as well as adding functionality that the scalability affords.**

I. INTRODUCTION

The *Ocean Observatories Initiative (OOI)* [1] is an NSF-funded program to establish the ocean observing infrastructure of the 21st century benefiting research and education. It will start its 5-year construction period in September 2009, promising to deliver cyber and physical observatory infrastructure components as well as substantial core instrumentation to study a wide range of environmental processes during an operational period of 25 years or more.

The OOI comprises three types of interconnected observatories spanning global, regional and coastal scales. The global component addresses planetary-scale problems via a network of moored buoys linked to shore via satellite. A regional cabled observatory will 'wire' a single region in the Northeast Pacific Ocean with a high-speed optical and power grid. The coastal component of the OOI will expand existing coastal observing assets, providing extended opportunities to characterize the effects of high frequency forcing on the coastal environment.

The *OOI Cyberinfrastructure (CI)* [9] constitutes the integrating element that links and binds the three types of marine observatories and associated sensors into a coherent system-of-systems. From an operational view, the *Data Distribution Network* (Fig. 1) lies at the heart of the Cyberinfrastructure enabling a multitude of science and education applications, ranging from observational data product generation, to data analysis, numerical ocean modeling, visualization and semantically enabled data query and transformation. In addition, the CI fundamentally supports a class of applications exploiting the knowledge gained from analyzing observational data for objective-driven ocean observing applications, such as automatically triggered response to episodic environmental events and interactive instrument tasking and control.

The U.S. Department of Commerce through NOAA operates the *Integrated Ocean Observing System (IOOS)* providing ocean and coastal data in various formats, rates and scales on open oceans and coastal waters to scientists, managers, businesses, governments, and the public to support research and inform decision-making. The NOAA IOOS program initiated development of the *Data Integration Framework (DIF)* [7] to improve management and delivery of an initial subset of ocean observations with the expectation of achieving improvements in a select set of NOAA's decision-support tools. OOI and NOAA through DIF collaborate on an effort to inte-

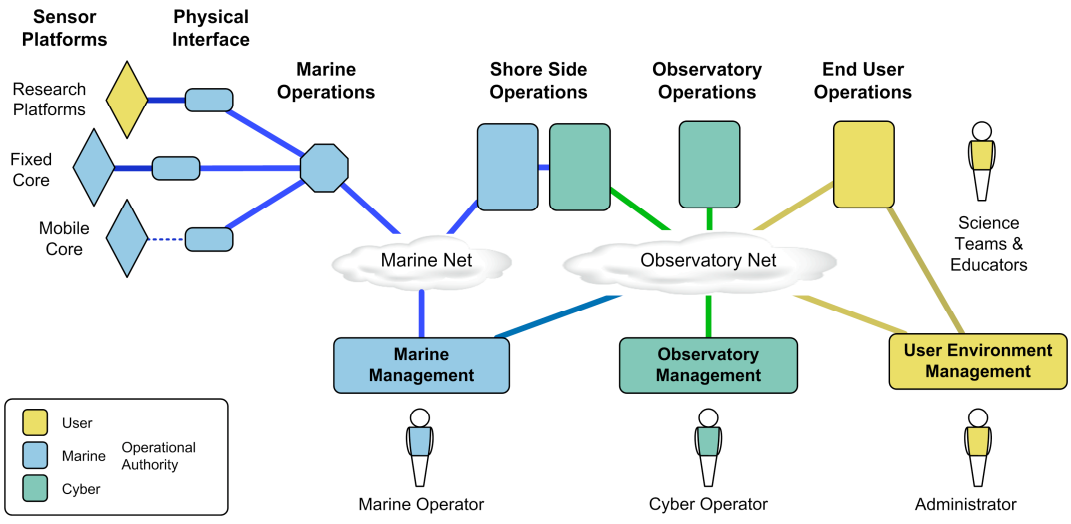

Fig. 1. OOI Data Distribution Network spanning multiple authority domains

0-933957-38-1 ©2009 MTS

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2010	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Serving Ocean Model Data on the Cloud	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at San Diego, La Jolla, CA**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM202806. Proceedings of the Oceans 2009 MTS/IEEE Conference held in Biloxi, Mississippi on 26-29 October 2009. U.S. Government or Federal Purpose Rights License, The original document contains color images.

14. ABSTRACT
The NOAA-led Integrated Ocean Observing system (IOOS) and the NSF-funded Ocean Observatories Initiative Cyberinfrastructure Project (OOI-CI) are collaborating on a prototype data delivery system for numerical model output

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**10**	

grate the data distribution, access and analysis needs of both programs.

Consider the example scenario from Fig. 2 with closed loop scientific investigation activities enabled by the OOI Integrated Observatory, based on CI software capabilities. Observational data from various sources are assimilated for numerical ocean modeling systems such as ROMS, the Regional Ocean Modeling System. The model outputs are then analyzed and evaluated, to be exploited with the purpose of refining future observations, for instance by providing specific tasking of observing programs using gliders or by reconfiguring sensors. The OOI CI provides the software services and user interfaces to support these applications; in addition it provides the underlying integration infrastructure consisting of message-based communication, governance and security frameworks, similar to the role of the operating system on a computer. The CI also provides the mechanisms to execute distributed processes anywhere in the network and connect then into a coherent system of systems.

In this paper, we report findings on several cloud-computing [1] based, data-distribution prototypes that will inform future developments of the OOI CI. These and further prototypes are currently being developed within the OOI pilot period, preceding the OOI construction. The purpose of all prototypes is risk mitigation, which includes technology maturity, integration complexity and community acceptance risk.

Specifically, the Unidata THREDDS Data Server and NOAA Environmental Research Division's Data Access Program (ERDDAP) have been deployed on Amazon's Elastic Compute Cloud (EC2), and used to deliver data from several different oceanographic models to scientists. Using on-demand elastically provisioned cloud compute and storage resources, the most recent prototype also provides server-side dataset caching and manipulation of structured and unstructured gridded data. A message-passing middleware connects the distributed processes within the cloud. This set of technologies realizes a major part of the technology backbone of the upcoming transformative OOI data distribution infrastructure and brings important new capabilities to IOOS data management. In this capacity, the combined resources enable science applications that go substantially beyond what is possible today.

Section II provides an overview of the OOI Cyberinfrastructure, Section III provides an overview of the NOAA DIF and Section IV presents background for the targeted technologies. Sections V and VI present the two prototypes we developed to evaluate the presented technologies and strategies for scaling them on Amazon's EC2 computing cloud infrastructure; finally, Section VII presents our findings and lessons learned.

[1] dynamically scalable collection of resources, most often computation and storage, provided as a service over the Internet

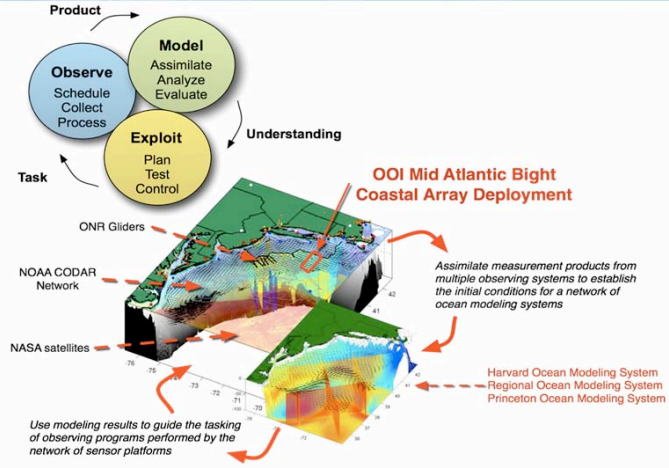

Fig. 2. Closed loop scientific investigation activities supported by the OOI Cyberinfrastructure

II. OOI CYBERINFRASTRUCTURE OVERVIEW

The core capabilities and the principal objectives of ocean observatories are collecting real-time data, analyzing data and modeling the ocean on multiple scales and enabling adaptive experimentation within the ocean. A traditional data-centric cyberinfrastructure, in which a central data management system ingests data and serves them to users on a query basis, is not sufficient to accomplish the range of tasks ocean scientists will engage in when the OOI [1] is implemented.

Instead, a highly distributed set of capabilities are required that facilitate: (1) end-to-end data preservation and access; end-to-end, (2) human-to-machine and machine-to-machine control of how data are collected and analyzed, direct, closed loop interaction of models with the data acquisition process; (3) virtual collaborations created on demand to drive data-model coupling and share ocean observatory resources (e.g., instruments, networks, computing, storage and workflows); (4) end-to-end preservation of the ocean observatory process and its outcomes; and (5) automation of the planning and prosecution of observational programs. An education component will provide an infrastructure for education and public engagement applications.

Hence, it is most appropriate to view the OOI as a whole—the *Integrated Observatory*—that through its CI component will allow scientists and citizens from a variety of authority domains (see Fig. 1) to view particular phenomena irrespective of the observing elements (e.g. coastal, global, regional, ships, satellites, IOOS…) to which the observations belong.

High-Level Architecture

The OOI CI provides a heterogeneous and large set of application and infrastructure services. To manage the complexities represented by these services, the CI is structured into six subsystems (see Fig. 3) partitioning the services into six services networks (SNs) that are interdependent. The application-supporting subsystems, described in [17], are Sensing and

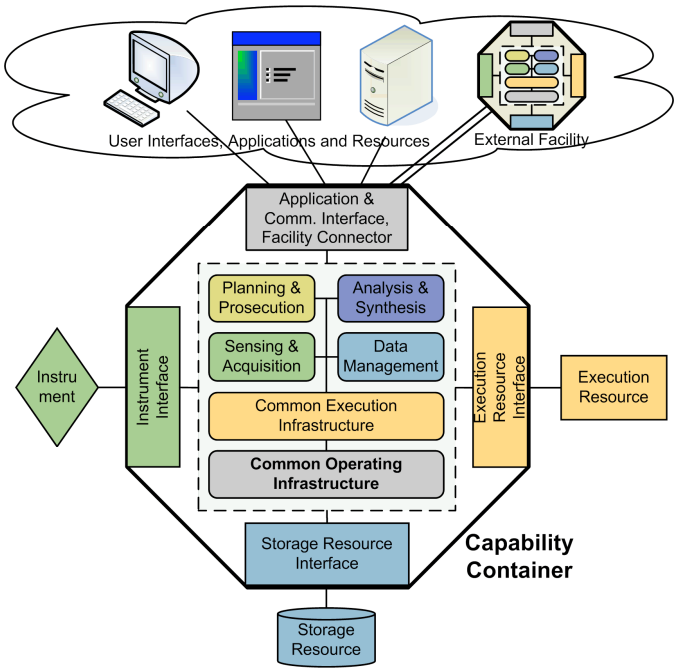

Fig. 3. OOI CI subsystems and interfaces, bundled as the CI capability container

Acquisition, Data Management, Analysis and Synthesis, and Planning and Prosecution. The infrastructure subsystems, described in [18], are Data Management, Common Execution Infrastructure (CEI), and Common Operating Infrastructure (COI). The Data Management subsystem provides both application and infrastructure services.

All subsystem services with their internal and external interfaces form the CI capability container, as shown in Fig. 3. Such capability containers can be deployed (i.e., installed, configured and connected to the rest of the network) where needed within the OOI Integrated Observatory network. Capability containers provide transparent and uniform network access to connected resources and provide infrastructure for system operation and management. Not all services have to be enabled or deployed with each capability container; this depends on host environment resource availability and required service capabilities. The CI capability container will be deployed, in particular, at the facilities participating in the OOI program forming the so-called CyberPoPs (Cyberinfrastructure Point of Presence).

Data Management Subsystem

The services provided by the Data Management subsystem (Fig. 4) support a wide range of scientific data and information management services; both supporting data-oriented applications and providing core infrastructure. Services include observational and external data ingestion into CI managed data repositories along with any associated metadata. Services also provide syntactical data format transformations as well as ontology-based data mediation, providing semantically enabled access to any data in repositories based on metadata as part of user provided search criteria.

From the point of view of science applications, the critical services are ingestion and transformation. The *Ingestion* service is responsible for initial data parsing, initial metadata extraction, registration, and versioning of data products received. The *Transformation* service manages the data content format conversion/transformation, mediation between syntax and semantics of data (based on ontologies), basic data calibration and QA/QC, additional metadata extraction, qualification, verification and validation. The *Presentation* service enables data discovery, access, reporting, and branding of data products. For data discovery, it provides the mechanisms to both browse/navigate specific data products and search/query of them based on specific metadata or data content.

At the infrastructure level, there are services that provide data distribution, preservation and inventory capabilities (Fig. 4). The Data Distribution Network—the Exchange—represents the main Data Management infrastructure element that enables the science data distribution throughout the Integrated Observatory network. It represents the integrating data distribution infrastructure that most of the Data Management services are relying on; it itself relies significantly on services from the Common Operating Infrastructure, in particular the *Messaging Service*, as described in the next section and in [4].

The infrastructure services provide information distribution, persistence and access across space and time, making data obtainable across the observatory network. Information includes observational data and derived data products along with other information resources required for the operation of the

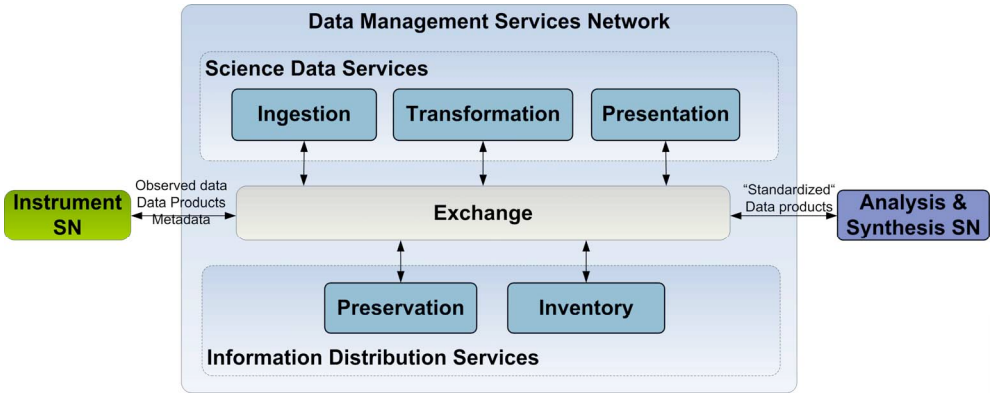

Fig. 4. OOI CI Data Management subsystem high-level service architecture

integrated observatory, such as ancillary instrument information, user identities, workflow definitions, and executable virtual machine images. The *Preservation* service is responsible for data replication, preservation, and archival/backup as defined by OOI policies. It also provides distributed data repository capabilities based on the underlying services of the COI subsystem. The *Inventory* service provides the cataloging, indexing and metadata management capabilities required for data ingestion and retrieval.

Message-Based Communication and Integration

The CI integration strategy defines how individual software components integrate into the system-of-systems through a message-broker integration infrastructure. The communication system of the OOI CI applies messaging as the central paradigm of inter-application information exchange, realizing the *Messaging Service*, the integrating element of all services. It is part of the *Common Operating Infrastructure (COI)*, the subsystem that provides the full set of integration frameworks and services (see [18], [4]).

Message-oriented middlewares (MOM) (see [6], [11]) are based on the concept of a message as the exclusive means of information exchange between the distributed components of a system. All information that is passed between two components or services is contained in messages exchanged asynchronously (i.e. non-blocking) over a communication infrastructure. The sender of a message does not wait for the message to be delivered or returned; it only waits for the MOM to acknowledge receipt of the message. Delivering messages to recipients utilizes the concept of queues. An application component in a message-oriented architecture only knows the incoming queues that it receives messages from as well as the outgoing queues it delivers messages to, plus the message formats that pertain to these queues. The MOM provides the capability for system integrators to connect these queues to known endpoints (i.e. addresses) in the network; consequently it handles routing, reliable storage and delivery of messages to intended recipients across the network. Standardization is on the way for the underlying message wire transport protocol: the Advanced Message Queuing Protocol (AMQP) [2] defines the interactions of a message broker with its clients, promising interoperability between message brokers of different provenance.

The left part of Fig. 5 depicts the fundamentals of the CI Messaging Service. Message brokers are the central infrastructure elements, represented as Exchange Points to all clients, responsible for the routing and delivery of messages. Message Clients provide the interfaces to the application logic. The right part of Fig. 5 provides an exemplar application scenario within the OOI CI. An Instrument Agent publishes a raw data stream on an Exchange Point (a queue) via messaging. Any number of consumers may choose to subscribe to such an exchange point. In this example, the data processing application as well as the data repository will receive the published messages. A data stream is a continuous series of related self-contained messages on a given exchange point. There is a second exchange point for another data product containing processed data that is consumed by an event detector process. The physical deployment of all applications is irrelevant as the Exchange abstracts and realizes all necessary connectivity.

Scalable Cloud-Enabled Infrastructure

In order to realize a scalable deployment of capability containers across the OOI network, the CI applies virtualization of computation and storage. As mentioned above, the capability containers from all facilities are interconnected and aggregated to present themselves as a coherent OOI integrated observatory. All resources are operated within their respective domains of authority, and policies specific to these facilities

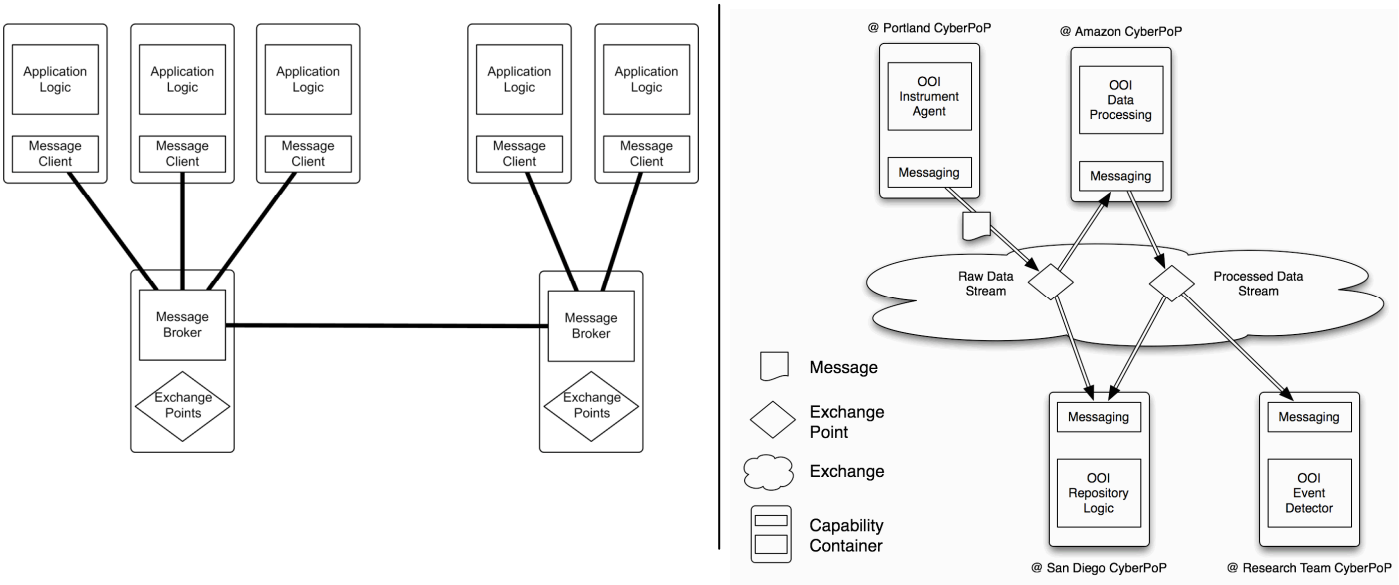

Fig. 5. OOI CI Message-broker architecture (left) and messaging scenario (right)

apply independent of on whose behalf the resources are used.

At the deployment level, implemented and integrated software components are bundled according to specific functional needs into deployable types (e.g., templates for virtual machine images) that can operate on dynamically instantiated virtual computation units (e.g., virtual machines). The CI manages a repository of such deployable types. In addition, it provides adapters for supported target execution environments from various facilities. These adapters turn deployable types into deployable units that are specific to one supported target environment. These units can be provisioned as needed and will register themselves in the operational environment where they are deployed (contextualization). Through these steps, the CI achieves independence of software components from specific target environments. In addition, it can react dynamically to increased resource needs or failures, and bring up new instances (elastic computing).

The described deployment mechanism reaches its full power in combination with cloud computing [12], where virtualized compute and storage resources are provided over the Internet. Departing from the classical grid strategy popular for high-performance computing applications, the CI employs a more flexible cloud-based strategy, where resources can be allocated on demand based on the actual needs and capabilities. The CEI subsystems implements the above described mechanisms for bundling software components, adapting these bundles to the target environment (e.g. Amazon's EC2 cloud [3]), provisioning such "images" as needed and contextualizing them within an operational context. The CEI will also schedule computation and required storage as needed and provision cloud resources that can match user requests. The entire process occurs automatically according to policy defined by the OOI CI operators (see details in [18]).

III. NOAA IOOS DIF Overview

The NOAA Integrated Ocean Observing System (IOOS) Data Integration Framework (DIF) (see [7], [8]) has established an experimental deployment of standardized information access services at several data providers of interest to NOAA (Fig. 6). The services include OPeNDAP Data Access Protocol (DAP), Open Geospatial Consortium (OGC) Sensor Observation Service (SOS), and OGC Web Map Service (WMS). The initial scope of the three year (2007-10) DIF project was limited to three NOAA data providers, four NOAA data customers (decision support tools), and seven observed oceanographic or meteorological quantities. The scope includes both *in situ* observations, remote-sensing observations, and ocean or coastal model data. The DIF scope is limited to data access from data centers, and does not include the transmission of data from the marine observing platforms to the data center.

IOOS DIF is now in an evaluation and expansion phase. Specifically, additional services beyond basic data access are being considered to enable other customers and providers to participate in IOOS. Two such services are being addressed by the OOI prototypes described here: service gateways, translate between different service interfaces (e.g., between DAP and SOS), and format conversion services. Additional opportunities to demonstrate the capacity of the emerging OOI CI to support operational IOOS data delivery requirements are anticipated (Fig. 6). IOOS Regions have also begun to collaborate with similar services. NDBC – NOAA National Data Buoy Center, CO-OPS – NOAA Center for Operational Oceanographic Products and Services.

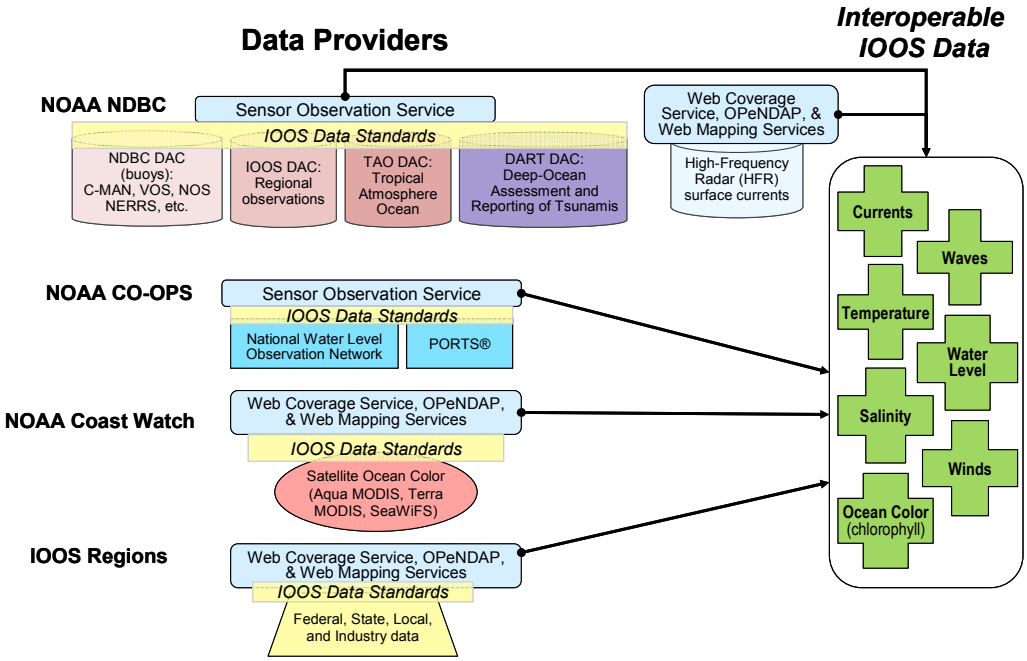

Fig. 6. Preliminary instantiation of DIF with three NOAA data providers

IV. TARGETED TECHNOLOGIES

For the purposes of the prototypes described in this paper, we have identified a set of key technologies that are relevant for an interoperable and flexible science data distribution infrastructure. Several of these technologies were already being used effectively by the NOAA DIF project, while others promised to extend the capabilities in important ways. In the following, we will introduce these technologies and their importance for the user community as well as for an integrated infrastructure.

OPeNDAP Data Access Protocol (DAP) is a protocol/service for accessing data from large scientific datasets over the Internet (see [10], [21]). Datasets can be queried to return the data types, sizes and attributes of the variables they contain, and then just selected variables and subsets of data can be returned. It was originally designed to work with large amounts of remote sensing data efficiently over the internet, and has been used heavily in the meteorology and oceanographic communities for the last 15 years. The DAP service has been implemented in numerous data delivery systems (e.g. GRADS Data Server, Hyrax Server, PyDAP, THREDDS Data Server), in leading scientific analysis environments (Python, Matlab, IDL, FERRET, GRADS) as well as other applications (e.g. Unidata Integrated Data Viewer, Panoply, ncWMS, ERDDAP) .

ERDDAP [15] was built by NOAA ERD to solve some very practical problems: How to translate data between a variety of scientific data formats and services (e.g. NetCDF files, DAP services, Sensor Observation Service). ERDDAP is a RESTful service that provides a simple, consistent way to access data from a variety of data transport services (e.g. DAP, DiGIR, SOS) and return the data in formats that are used in real applications (e.g. .mat, .csv,.kml. .nc ,.png), including web development formats (e.g .json, geojson, .htmlTable) ERDDAP also acts as a DAP, SOS and Web Mapping Service (WMS) server, so that a DAP, SOS or WMS client can access data using these protocols from any of the data sources ERDDAP knows about, even if that data source does not support those protocols. Thus, using ERDDAP, users can, for example, request a subset of gridded data from a DAP server and deliver a NetCDF file or request time series data via SOS and deliver an Excel Spreadsheet. Along with the requested data or plot, users are supplied the URL that accomplishes the requested task, so that they can easily create scripting applications or powerful processing chains. Gridded data must have uniformly spaced geographic coordinates to work with ERDDAP.

THREDDS Data Server (TDS) by Unidata [22] is a Tomcat application that allows a variety of data from a variety of data formats and services to be served via DAP, the OGC Web Coverage Service and the OGC Web Map Service. It supports serving HDF4, HDF5, GRIB, GRIB2, NetCDF3 and NetCDF4 files, as well as virtual datasets constructed with the NetCDF Markup Language (NcML). NcML allows non-standard datasets to be standardized through the addition or correction of metadata as well as a variety of aggregation capabilities. The Ferret THREDDS Data Server (F-TDS) [16] is a customized version of Unidata's THREDDS Data Server; it provides server-side functions such as time averaging and regridding to be specified in the URL. Using the F-TDS, for instance, a user who accesses a dataset of daily temperature data who wants only the annual mean can specify this calculation on the URL, saving tremendous bandwidth but placing additional computation demands on the server.

GridFields [13] is a library for the algebraic manipulation of scientific datasets. This includes in particular unstructured gridded datasets. Unstructured grids use a mesh of polygons to describe complex structures. One of the benefits of using polygons is the flexibility of resolution and topology. Following a naturally irregular coastline or riverbed benefits strongly from unstructured gridded data representations. In structured grids, much of the resolution or space would be lost to areas of little interest. One of the downsides of unstructured gridded data is the difficulty to work with it, such as subsetting. GridFields provides the library to do so more efficiently

Amazon Web Services – starting in early 2006, Amazon provided a set of computational and storage capabilities for third party applications [3]. The mechanisms for provisioning and controlling these resources rely on standard Web Services technologies. The major advantage compared with traditional solutions is that cost depends on the actual usage instead of high upfront costs for hardware that might not be used all time. Additional benefits come from the reliability and redundancy of the provided solution and the on-demand scaling with user/application demands. It also allows for rapid prototyping of new technologies with full flexibility in choosing the right architecture for the target application domain.

V. PROTOTYPE 1: ERDDAP ON AMAZON EC2

A first prototype was developed in late 2008 providing web-based user access to oceanographic data through NOAA's ERDDAP, as well as efficient science metadata caching, and scalable deployment in a cloud computing environment using Amazon's EC2. Fig. 7 depicts the architecture of this prototype [19].

The central functions of the prototype are provided by ERDDAP (depicted as ERDDAP Utility) as described above. ERDDAP is a server-hosted Java application that provides a web interface for registering data sources (datasets) that are available via HTTP through the Internet. It also provides a web interface to list all available data sources, show their detailed metadata and to query for specific datasets and specific variables. Data consumers can also filter the data sources to the subsets of interest and retrieve the available data sources in a number of formats that ERDDAP can provide. This includes DAP datasets and visualization graphs.

ERDDAP is designed as a single server application. There exists no scaling strategy for scenarios where the user requested transformations and the resulting server load exceeds the capabilities of the server. Although it is possible to add

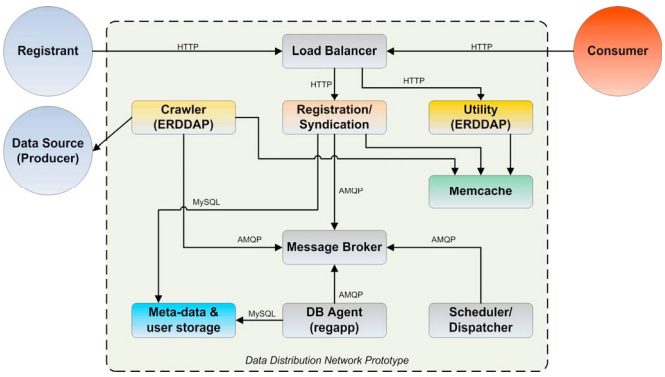

Fig. 7. ERDDAP cloud prototype architecture

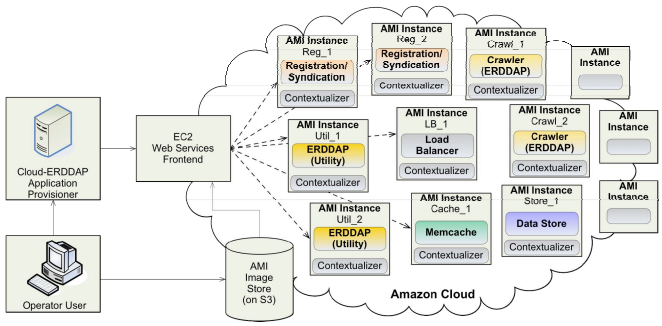

Fig. 8. ERDDAP cloud prototype deployment scenario

multiple ERDDAP servers behind a load-balancer, this option has the disadvantage of causing higher load on the data sources that are repeatedly queried for updates and may lead to potential inconsistent state between all ERDDAP instances at one given instant, leading to a confusing experience for the users.

Even if multiple instances can be provided, further scalability problems arise eventually through the limitation of computational resources at the site where the ERDDAP application is running. Typically, the hardware environment, such as a grid cluster, is designed with a specific load estimate and has a predetermined capacity. Additional computations cannot be sustained.

The OOI strategy is to use cloud resources to deploy all components of the architecture as a service. This strategy enables the allocation of computational 'instances' on the cloud on demand. For instance, if we detect that ERDDAP utility servers are under high load, we can add another instance to the existing cluster with little or no disruption to the existing system. We designed the prototype such to allow adding and removing instances of each component flexibly when needed.

In addition, we designed the components such that their load characteristics are optimally reflected by our scaling strategy. For instance, we split the ERDDAP application into three different types of processes: (1) The ERDDAP utility that provides the web interfaces and the transformation engine, (2) the ERDDAP crawler, which regularly queries the data sources for updates of data and metadata, and (3) the Memcache component that provides the distributed shared state between all instances efficiently. The Metadata and user storage component was realized as a MySQL cluster, allowing for multiple instances of MySQL engine and storage processes. Other types of processes, such as a scheduler, the software load-balancer, and the message broker completed the architecture depicted in Fig. 7.

Using the cloud also relieves us of common operations concerns such as power and cooling management, hardware failure, personnel support and network setup. Fig. 8 shows the deployment view on the prototype, with components in our local hardware environment, with virtual machine images within Amazon's S3 storage cloud, and running instances connected by the message-broker infrastructure on the EC2 cloud.

A local provisioner process is responsible for starting and controlling the entire system. Startup of a typical configuration consisting of 15 different virtual machine images takes less than 2 minutes as part of a fully automated process. A single web URL provided the users' entry point to the system.

VI. PROTOTYPE 2: DATA EXCHANGE ON EC2

The primary objective of the second prototype described in this paper is to deploy a scalable "Data Exchange" infrastructure, leveraging the findings and technologies of the first prototype described in the previous section [20]. The Data Exchange prototype, currently in development, will provide a server-side data processing capability for use by an initial set of active ocean modeling communities to efficiently exchange large model datasets in whole or in part while preserving the original content and structure of the dataset. The targeted modeling communities participating in this effort include NERACOOS, MARCOOS and SCCOOS. In a subsequent phase the Data Exchange will be promoted for broader use by the IOOS community.

Arising from this primary objective, the effort will provide the IOOS DIF with a platform on which to test the Web Services and Data Encoding being developed to distribute the seven "Core Variables" (Currents, Water Level, Sea Temperature, Salinity/Conductivity, Surface Winds, Waves and Chlorophyll) to their four initial target groups. The process of developing and deploying the Data Exchange in the context of operational user communities will drive the refinement of the OOI Cyberinfrastructure requirements, design and technology choices prior to the start of construction of the OOI. Finally this effort will also provide the IOOS DMAC with practical insight into viable strategies for realizing an integrated national ocean observing cyberinfrastructure.

The Data Exchange's central premise is to provide modeling communities with an effective community infrastructure to publish datasets and server-side functions that:

1. Register and transmit Datasets of any structural type supported by DAP,
2. Register and transmit Virtual Datasets authored using NcML,
3. Register and execute "Ferret" conformant server-side Functions,
4. Register and trigger Subscriptions that follow the evolution of a Dataset,
5. Register and execute "Data Exchange" conformant Tasks (such as server-side analyses),
6. Link data subscription notifications to the execution of a "Data Exchange" Task,
7. Register and manage "Data Exchange" Communities to delineate and control access to Datasets, Functions, Subscriptions and Tasks.

The Data Exchange as community infrastructure will focus on the following core concerns:

1. Transparency - Existing publishers and consumers will be able to use the infrastructure without making changes to their current practices and processes,
2. Elasticity - The infrastructure will automatically adjust its computing and storage capacity to meet demand
3. Fault-Tolerance – The infrastructure will continue to operate and self-heal in the presence of any infrastructure component failure; i.e., network, storage, computer and/or process.

At the logical level, the Data Exchange prototype provides a collection of services for Data Retrieval, Operations and Management, Front-end capabilities, Data and Metadata Caching with Storage and Retrieval Capabilities (see Fig. 10). There

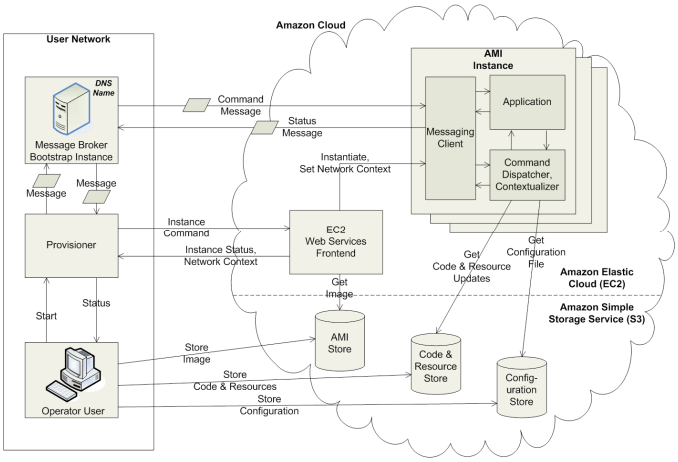

Fig. 9. Prototype EC2 cloud provisioning deployment strategy

could be multiple instances of a given service at any moment in time. For instance, consider multiple Fetchers implementing transparent access mechanisms to external data sources and scaling with the demand. The Data Exchange relies on a number of infrastructure capabilities such as the Messaging Service, the Storage Repository, and the Attribute Store, which are abstract services provided as standalone components for reuse purposes. Fault tolerance is provided both through replication of necessary capabilities under the control of the Resource Manager and Scheduler, and underlying reliable messaging capabilities of the Messaging Service.

The Data Exchange prototype builds on the cloud provisioning infrastructure of the ERDDAP prototype as described in the previous section. It makes use of an AMQP based message-passing middleware that provides reliable communica-

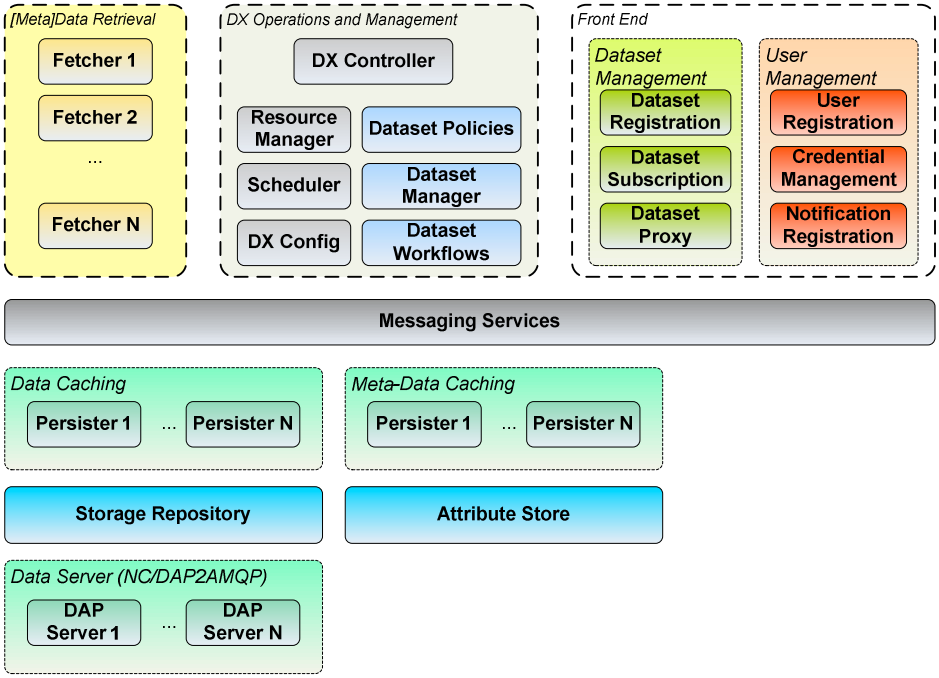

Fig. 10. Data Exchange prototype logical architecture

tion between system components The Data Exchange includes a Ferret THREDDS server as part of a distributed architecture deployed on the Amazon EC2 cloud, as depicted in Fig. 9. The web frontend and catalog components are separated from the server-side dataset caching and indexing components as well as from server-side processing functions. Users can apply Ferret functions to manipulate cloud-cached model data structured in form of rectilinear and curvilinear grids, for instance to subset and time average. A wide range of operations and grid types (including unstructured) is supported though server-side function via the GridFields library [13].

Users can access the prototype using their Matlab environments and via the web browser. Matlab integration with the Data Exchange prototype is provided via the Matlab Structs Tool [14]. This provides access to any DAP-accessible dataset, which includes Ferret results. A small amount of Matlab code provides a simple interface for creating server-side Ferret programs from within Matlab scripts.

Based on the F-TDS example, we plan to create a similar TDS/GridFields plugin and corresponding Matlab code, so that scientists can more easily work with remote datasets including unstructured grids.

We are also investigating how to add Data Exchange support to existing systems. SOCKS or HTTP proxies, modified DAP libraries are both possible and being considered. Proxies would allow the flexible deployment of cloud-based and local caches without changing working programs.

VII. Results and Lessons Learned

The separation of storage (caching) and processing components within the architecture enables automatic dynamic provisioning of cloud resources based on demand. When users register new datasets that need to be cached within the cloud infrastructure, new storage resources are elastically provisioned within seconds contingent on policy. Similarly, as new requests for server-side processing arrive from users, potentially requiring significant amount of CPU cycles, new virtual servers are provisioned within seconds and removed once processing completes.

The ability to split the ERDDAP application into several independently-scalable pieces, and the relative inability of some other software packages to be similarly decomposed, clearly indicates the importance of software factorization during development, and of considering what factorization choices are best to enable independent modules to either scale independently or be coupled in a cloud-computing environment.

The results from the ERDDAP and Data Exchange prototypes are very encouraging and valuable. Through the application of cloud provisioning technologies it is possible to deploy larger-scale distributed applications with little effort within a short time; such applications show a high reliability and resilience to failure and they can adapt to load and demand elastically. OOI is currently implementing a further advanced prototype for elastic scaling of distributed applications on demand using resources from multiple clouds, leading to a comprehensive cloud execution infrastructure, the technological basis for the future OOI CI Common Execution Infrastructure subsystem.

Providing a robust data distribution infrastructure is of substantial value for the ocean modeling communities. The availability of an easy to use, reliable and flexible community infrastructure that goes beyond current capabilities provides immediate benefit to groups with fewer available resources; it also provide infrastructure operators and sponsors with the possibility to apply economies of scale to a central infrastructure component, leading to lower cost of operation and optimized resource utilization. In the mid-term, such an infrastructure also increases the drive towards stronger interoperability of data distribution technologies and data representation formats.

For the OOI, an early adoption of a community infrastructure based on the prototyped technologies is of high value, because it promises to lead to earlier and larger community acceptance of future OOI CI infrastructure elements and technologies. In the short-term, the experiences gained with the described prototypes are an effective means for technology risk mitigation prior to the OOI construction period.

IOOS is particularly interested in elements of the NSF OOI CI research that can be used in an operational context as early as possible. The success of the cloud deployment prototyping suggests that popular ocean data providers should transmit a copy of their data to the cloud for general access and distribution, while retaining the master copy locally for limited access. The authors anticipate expanding the OOI CI and IOOS collaboration via a series of demonstration activities as their mutual activities mature. The benefits of such a collaboration based on the described technologies promise to extend well beyond the scope of the DIF.

Acknowledgment

The prototype effort articulated in this paper is funded through the NOAA cooperative agreement, # NA17RJ1231, with the Scripps Institution of Oceanography University of California, San Diego. The Data Exchange effort is a part of and dependent on the larger OOI Cyberinfrastructure effort funded through the JOI Subaward, JSA 7-11, which is in turn funded by the NSF contract OCE-0418967 with the Consortium for Ocean Leadership, Inc.

References

[1] Ocean Observatories Initiative (OOI). Program website, http://www.oceanleadership.org/ocean_observing/ooi

[2] Advanced Message Queuing Protocol (AMQP). AMQP Working Group Website http://www.amqp.org/

[3] Amazon.com, Amazon Web Services for the Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/.

[4] M. Arrott, A.D. Chave, C. Farcas, E. Farcas, J.E. Kleinert, I. Krueger, M. Meisinger, J.A. Orcutt, C. Peach, O. Schofield, M. Singh, F.L. Vernon. Integrating Marine Observatories into a System-of-Systems: Messaging in the US Ocean Observatories Initiative. In Proc. MTS/IEEE Oceans

2009 Conf, IEEE Marine Technical Society, paper #090601-019- (this volume), Oct. 2009.
[5] M. Arrott, B. Demchak, V. Ermagan, C. Farcas, E. Farcas, I. H. Krüger, M. Menarini. Rich Services: The Integration Piece of the SOA Puzzle. In Proc. of the IEEE International Conference on Web Services (ICWS), Salt Lake City, Utah, USA. IEEE, Jul. 2007, pp. 176-183.
[6] G. Banavar, T. Chandra, R. Strom and D. Sturman. A case for message oriented middleware. Proc. of the 13th International Symposium on Distributed Computing, pp. 1–18, 1999.
[7] J. de La Beaujardiere. The NOAA IOOS Data Integration Framework: Initial Implementation Report. In Proc. MTS/IEEE Oceans 2008 Conf., IEEE Marine Technical Society, paper #080515-116, Sept. 2008.
[8] J. de La Beaujardiere. IOOS Data Management Activities. In Proc. MTS/IEEE Oceans 2009 Conf, IEEE Marine Technical Society, paper #090529-023 (this volume), Sept. 2009.
[9] A. Chave, M. Arrott, C. Farcas, E. Farcas, I. Krueger, M. Meisinger, J. Orcutt, F. Vernon, C. Peach, O. Schofield, and J. Kleinert. Cyberinfrastructure for the US Ocean Observatories Initiative: Enabling Interactive Observation in the Ocean. In Proc. IEEE OCEANS'09 Bremen, Germany. IEEE Ocean Engineering Society, May 2009.
[10] P. Cornillon, J. Gallagher, T. Sgouros. OPeNDAP: Accessing data in a distributed, heterogeneous environment. Data Science Journal vol. 2, pp. 164-174, 2003.
[11] P.T. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. Tech. Rep. DSC ID:2000104, EPFL, January 2001.
[12] B. Hayes. Cloud Computing. Comm. ACM, 51(7):9–11, 2008.
[13] B. Howe. GridFields: Model-Driven Data Transformation in the Physical Sciences. Phd Dissertation, Portland State University, 2007.
[14] Matlab Structs Tool (loaddap), Website http://opendap.org/download/ml-structs.html
[15] NOAA ERDDAP. Website http://coastwatch.pfeg.noaa.gov/erddap/
[16] NOAA PMEL Ferrets-THREDDS Data Server (F-TDS). Website http://ferret.pmel.noaa.gov/LAS/documentation/the-ferret-thredds-data-server-f-tds/
[17] OOI CI Integrated Observatory Applications Architecture Document, OOI controlled document 2130-00001, version 1-00, 10/28/2008, available at http://www.oceanobservatories.org/spaces/display/FDR/CI+Technical+File+Repository
[18] OOI CI Integrated Observatory Infrastructure Architecture Document, OOI controlled document 2130-00002, version 1-00, 10/24/2008, available at http://www.oceanobservatories.org/spaces/display/FDR/CI+Technical+File+Repository
[19] OOI CI Data Distribution Network Prototype. http://www.oceanobservatories.org/spaces/display/CIDev/Data+Distribution+Network
[20] OOI CI Data Exchange Prototype. http://www.oceanobservatories.org/spaces/display/CIDev/Data+Exchange
[21] OPeNDAP.org. OPeNDAP Framework. Website http://opendap.org/
[22] Unidata THREDDS Data Server (TDS). Website http://www.unidata.ucar.edu/projects/THREDDS/